神经离散表示学习

Aaron van den Oord
DeepMind
avdnoord@google.com

Oriol Vinyals
DeepMind
vinyals@google.com

Koray Kavukcuoglu
DeepMind
korayk@google.com

摘 要

在没有监督的情况下学习有用的表示仍然是机器学习的关键挑战。在本文中，我们提出了一个简单而强大的生成模型来学习这种离散表示。我们的模型，即矢量量化变分自动编码器（VQ-VAE），与 VAE 的不同之处在于两个关键方面：编码器网络输出离散码而不是连续码;并且先验是学习的而不是静态设置的。为了学习离散的潜在表示，我们结合了矢量量化（VQ）的思想。使用 VQ 方法允许模型避开“posterior collapse”（后验坍塌）问题 - 当潜编码与强大的自回归解码器配对时忽略潜编码 - 通常在 VAE 框架中观察到。将这些表示与自回归先验配对，该模型可以生成高质量的图像，视频和语音，以及进行高质量的说话者转换和音素的无监督学习，提供了学习表示具有实用性的进一步证据。

1 介绍

图像[38,12,13,22,10]，音频[37,26]和视频[20,11]的生成建模的最新进展已经产生了令人印象深刻的样本和应用[24,18]。与此同时，诸如少样本学习[34]，领域适应[17]或强化学习[35]等具有挑战性的任务在很大程度上依赖于原始数据中的学习表示，但是以无人监督的方式训练的通用表示仍远未成为有用的主导方法。

最大似然和重建误差是用于训练像素域中的无监督模型的两个常见目标，但是它们的有用性取决于所使用的特征的特定应用。我们的目标是实现保留其中的数据在潜在空间的重要特征，同时优化最大似然度的模型。正如[7]中的工作所表明的那样，最好的生成模型（通过对数似然度测量）将是没有潜编码的但是具有强大的解码器（例如 PixelCNN）。然而，在本文中，我们主张学习离散和有用的潜编码变量，我们在各种领域进行了论证。

具有连续特征的表示学习一直是许多先前工作的焦点[16,39,6,9]，但是我们专注于离散表示[27,33,8,28]，它们可能更适合我们的许多模式。语言本质上是离散的，类似地，语音通常表示为一系列符号。图像通常可以用语言简洁地描述[40]。此外，离散表示非常适合复杂的推理，计划和预测学习（例如，如果下雨，我会使用一把伞）。虽然在深度学习中使用离散潜编码已经证明具有挑战性，但已经开发了强大的自回归模型用于在离散变量上建模分布[37]。

在我们的工作中，我们引入了一个新的生成模型家族，通过对给定观察的（离散）编码的后验分布的新参数化，成功地将变分自动编码器（VAE）框架与离散潜编码相结合。我们的模型依赖于矢量量化（VQ），训练简单，不会出现大的差异，并且避免了

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

“后验坍塌”问题这一对于许多具有强大解码器的 VAE 模型一直存在的问题，这些解码器的问题通常由对潜编码的忽略而引起。此外，它是第一个与其连续对应物具有相似性能的离散潜编码 VAE 模型，同时提供离散分布的灵活性。我们将我们的模型称为 VQ-VAE。

由于 VQ-VAE 可以有效利用潜在空间，因此它可以成功地模拟通常跨越数据空间中许多维度的重要特征（例如，对象跨越图像中的许多像素，语音中的音素，文本片段中的消息等）。而不是集中注意力或将能力放在通常是局部的噪声和难以察觉的细节上。

最后，一旦 VQ-VAE 发现了模态的良好离散潜在结构，我们就会在这些离散随机变量上训练强大的先验，产生有趣的样本和有用的应用。例如，当接受语音训练时，我们发现语言的潜在结构，而没有任何关于音素或单词的监督或先验知识。此外，我们可以为我们的解码器配备扬声器标识，这允许扬声器转换，即：将语音从一个扬声器传送到另一个扬声器而不改变内容。我们还展示了学习 RL 环境长期结构的有希望的结果。

因此，我们的贡献可归纳为：

- 引入简单的 VQ-VAE 模型，使用离散的潜编码，不会遭受“后验坍塌”且没有变化问题。
- 我们证明了离散潜在模型（VQ-VAE）在对数似然中的表现与其连续模型对应物一样。
- 当与强大的先验技术配合使用时，我们的样本在语音和视频生成等各种应用中具有连贯性和高质量。
- 我们通过原始语音展示学习语言的证据，没有任何监督，并展示无监督的说话人转换的应用。

2 相关工作

在这项工作中，我们提出了一种新的方法来训练变分自动编码器[23,32]与离散潜在变量[27]。在深度学习中使用离散变量已经证明具有挑战性，正如大多数当前工作中连续潜变量的主导地位所表明的那样 - 即使潜在的模态本质上是离散的。

存在许多用于训练离散 VAE 的替代方案。 NVIL [27]估计器使用单样本目标来优化变分下界，并使用各种方差减少技术来加速训练。 VIMCO [28]优化了多样本目标[5]，通过使用来自推理网络的多个样本，进一步加速了收敛。

最近，一些作者建议使用基于所谓的 Concrete [25]或 Gumbel-softmax [19]分布的新的连续再聚合，这是一种连续分布，并且具有可以在训练期间退火的温度常数以收敛到极限的离散分布。在训练开始时，梯度的方差很小但有偏差，并且在训练结束时方差变得很高但是没有偏差。

然而，上述方法都没有用连续潜变量来弥补 VAE 的性能差距，其中人们可以使用高斯重新参数化技巧，这得益于梯度的低得多的方差。此外，这些技术中的大多数通常在相对较小的数据集（例如 MNIST）上进行评估，并且潜在分布的维度很小（例如，低于 8）。在我们的工作中，我们使用三个复杂的图像数据集（CIFAR10，ImageNet 和 DeepMind Lab）和原始语音数据集（VCTK）。

我们的工作还扩展了研究领域，其中自回归分布用于 VAE 的解码器和/或之前的[14]。这已经用于使用 LSTM 解码器进行语言建模[4]，最近进行了扩张卷积解码器[42]。 PixelCNNs [29,38]是卷积自回归模型，它也被用作 VAE 解码器中的分布[15,7]。

最后，我们的方法还涉及使用神经网络进行图像压缩的工作。 Theis 等人 [36]在算术编码之前使用标量量化来压缩激活以进行有损图像压缩。其他作者[1]提出了一种具有矢量量化的类似压缩模型的方法。

作者提出了矢量量化的连续放宽，其随时间退火以获得硬聚类。 在他们的实验中，他们首先训练自动编码器，然后将矢量量化应用于编码器的激活，并且最后使用具有小学习速率的软到硬放松来微调整个网络。 在我们的实验中，我们无法从头开始使用软到硬松弛方法，因为解码器始终能够在训练期间反转连续放松，因此不会发生实际的量化。

3 VQ-VAE

也许与我们的方法最相关的工作是 VAE。 VAE 由以下部分组成：编码器网络，其参考给定输入数据 x 的离散潜在随机变量 z 的后验分布 q（z|x），先验分布 p（z）和具有分布 p（x|z）的解码器输入数据。

通常，假设 VAE 中的后验和先验通常是以对角线协方差分布的，这允许使用高斯重新参数化技巧[32,23]。 扩展包括自回归先验和后验模型[14]，归一化流[31,10]和逆自回归后验[22]。

在这项工作中，我们介绍了 VQ-VAE，我们使用离散潜在变量和一种新的训练方式，受矢量量化（VQ）的启发。 后验和先验分布是分类的，从这些分布中抽取的样本索引嵌入表。 然后将这些嵌入用作解码器网络的输入。

3.1 离散潜在变量

我们定义了潜在的嵌入空间$e \in R^{K\times D}$，其中 K 是离散潜在空间的大小（即 K 路分类），D 是每个潜在嵌入向量e_i的维数。注意，存在 K 个嵌入向量$e_i \in R^D$，i ∈1; 2; ... ; K。如图 1 所示，模型采用输入 x，输入 x 通过编码器产生输出$z_e(x)$。然后使用共享嵌入空间 e 通过最近邻查找来计算离散潜在变量 z，如等式 1 所示。解码器的输入是对应的嵌入向量e_k，如等式 2 中给出的。可以看到该 0 前向计算通道作为常规自动编码器，具有特定的非线性，将潜在编码映射到 1-K 嵌入向量。模型的完整参数集是编码器，解码器和嵌入空间 e 的参数的并集。为简单起见，我们使用单个随机变量 z 来表示本节中的离散潜在变量，但是对于语音，图像和视频，我们实际上分别提取了 1D，2D 和 3D 潜在特征空间。

后分类分布 q（z|x）概率定义为 one-hot，如下：

$$q(z=k|x)=\begin{cases}1 & \text{for } k=\operatorname{argmin}_j \|z_e(x)-e_j\|_2,\\ 0 & \text{otherwise}\end{cases}, \tag{1}$$

其中$z_e(x)$是编码器网络的输出。 我们将此模型视为 VAE，我们可以在其中将 log p（x）与 ELBO 绑定。 我们的建议分布 q（z = k|x）是确定性的，并且通过在 z 上定义简单的均匀先验，我们获得 KL 散度常数并且等于 log K.

表示$z_e(x)$通过离散化瓶颈，然后映射到嵌入 e 的最近元素，如等式 1 和 2 中给出的。

$$z_q(x)=e_k, \quad \text{where} \quad k=\operatorname{argmin}_j \|z_e(x)-e_j\|_2 \tag{2}$$

3.2 学习

注意，没有为等式 2 定义实际梯度，但是我们近似梯度类似于直通估计器[3]，只是将解码器输入$z_q(x)$的梯度复制到编码器输出$z_e(x)$。 也可以通过量化操作使用子梯度，但是这个简单的估计器对于本文的初始实验很有效。

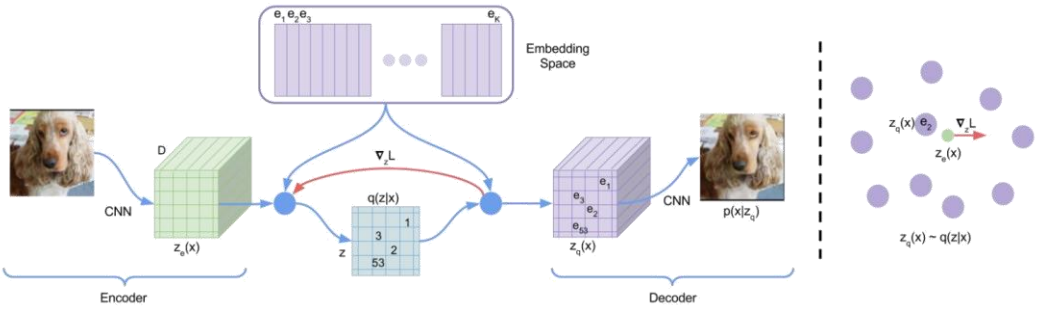

图 1：左图：描述 VQ-VAE 的图。 右图：嵌入空间的可视化。 编码器 z（x）的输出映射到最近的点 e_2。 梯度 $\nabla_z L$（红色）将推动编码器改变其输出，这可能会改变下一个正向传递中的配置。

在正向计算期间，最近的嵌入 $z_q(x)$（等式 2）被传递到解码器，并且在向后传递期间，梯度 $\nabla_z L$ 不加改变地传递到编码器。由于编码器的输出表示和解码器的输入共享相同的 D 维空间，因此梯度包含有关编码器如何改变其输出以降低重建损失的有用信息。

如图 1（右）所示，梯度可以推动编码器的输出在下一个正向传递中以不同方式离散，因为等式 1 中的分配将是不同的。

公式 3 规定了整体损失函数。它有三个组件用于训练 VQ-VAE 的不同部分。第一项是重构损失（或数据项），它优化了解码器和编码器（通过上面说明的估计器）。由于从 $z_e(x)$ 到 $z_q(x)$ 的映射的直通梯度估计，嵌入 e_i 不从重建损失 $\log p(z|z_q(x))$ 接收梯度。因此，为了学习嵌入空间，我们使用最简单的字典学习算法之一——矢量量化（VQ）。 VQ 目标使用 l2 误差将嵌入向量 e_i 移向编码器输出 $z_e(x)$，如公式 3 的第二项所示。因为该损失项仅用于更新字典，所以也可以更新字典项目作为 $z_e(x)$ 移动平均值的函数（不用于本工作中的实验）。有关详细信息，请参阅附录 A.1。

最后，由于嵌入空间的体积是无量纲的，如果嵌入 e_i 不像编码器参数那样快速地训练，则它可以任意增长。为了确保编码器提交嵌入并且其输出不增长，我们添加了承诺损失（commitment loss），即等式 3 中的第三项。因此，总训练目标变为：

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2, \tag{3}$$

其中 sg 代表在正向计算时被定义为标识的停止梯度运算符，并且具有零偏导数，因此有效地将其操作数约束为未更新的常量。解码器仅优化第一个损失项，编码器优化第一个和最后一个损失项，并且编码由中间损失项来优化。我们发现得到的算法结果对于不同β都非常稳健，因为对于范围从 0.1 到 2.0 的β值，结果没有变化。我们在所有实验中都使用β= 0.25，尽管这通常取决于重建损失的规模。由于我们假设 z 的统一先验，因此通常出现在 ELBO 中的 KL 项是常数（关于编码器参数）因此可以被忽略用于训练。

在我们的实验中，我们定义了 N 个离散的潜编码（例如，我们使用 ImageNet 的 32 x 32 潜编码，或 CIFAR10 的 8 x 8 x 10）。由此产生的损失 L 是相同的，除了我们得到 k-means 和承诺损失（commitment loss）的 N 个项上的平均值 – 对于每个潜编码中的一个。

完整模型 log p（x）的对数似然可以评估如下：

$$\log p(x) = \log \sum_k p(x|z_k)p(z_k),$$

因为解码器 p（x|z）是用来自 MAP 推理的 z = $z_q(x)$ 训练的，所以解码器一旦完全收敛，就不应该对于 z ≠ $z_q(x)$ 将任何概率权重分配给 p（x|z）。 因此，我们可以认为

$\log p(x) \approx \log p(x|z_q(x))p(z_q(x))$。 我们在第 4 节中凭经验评估这种近似。从 Jensen 的不等式，我们也可以写成 $\log p(x) \geq \log p(x|z_q(x))p(z_q(x))$。

3.3 先验

离散潜编码点 p（z）上的先验分布是分类分布，并且可以通过依赖于特征映射中的其他 z 来进行自回归。 在训练 VQ-VAE 的同时，先验保持恒定和均匀。 训练之后，我们在 z，p(z)上拟合自回归分布，这样我们就可以通过先前采样生成 x。 我们在图像的离散潜编码点上使用 PixelCNN，在原始音频上使用 WaveNet。 共同训练先验分布和 VQ-VAE，这可以加强我们的成果，留作未来的研究。

4 实验

4.1 与连续变量的比较

作为第一个实验，我们将 VQ-VAE 与正常 VAE（具有连续变量）以及具有独立高斯或分类先验的 VIMCO [28]进行比较。我们在 CIFAR10 上使用相同的标准 VAE 架构训练这些模型，同时改变潜空间容量（连续或离散潜在变量的数量，以及离散空间 K 的维数）。编码器由 2 个跨步卷积层组成，步长为 2，窗口大小为 4×4，后面是两个 3×3 残余块（实现为 ReLU，3x3 卷积，ReLU，1x1 卷积），全部具有 256 个隐藏单元。解码器类似地具有两个 3×3 残余块，接着是两个具有步幅 2 和窗口大小 4 的转置卷积。我们使用具有学习速率 2e-4 的 ADAM 优化器[21]并且在 250,000 步骤之后对于 VIMCO 评估批量大小为 128 的性能，我们在多样本训练目标中使用 50 个样本。

VAE，VQ-VAE 和 VIMCO 模型分别获得 4.51 位/ dim，4.67 位/ dim 和 5.14。所有报告的似然度都是下限。我们的连续 VAE 数据与深度卷积 VAE 报告的数据相当：此数据集上的数据为 4.54 位/ dim [13]。

我们的模型是第一个使用离散潜在变量的模型，它们挑战连续 VAE 的性能。因此，我们得到非常好的重建，如常规 VAE 提供的一样，具有符号表示提供的压缩表示。我们训练的 VQ-VAE 的一些有趣特征，含义和应用将在下一小节中介绍。

4.2 图像

图像包含大量冗余信息，因为大多数像素是相关且有噪声的，因此像素级的学习模型可能是浪费的。

在这个实验中，我们表明我们可以通过纯粹的去卷积 p（x|z）将它们压缩到 z = 32×32×1 的离散空间（K = 512）来模拟 x = 128×128×3 的图像。 因此减少 $\frac{128\times128\times3\times8}{32\times32\times9} \approx 42.6$位。 我们通过在 z 上学习强大的先验（PixelCNN）来模拟图像。 这 32×32×9 不仅可以大大加快训练和采样速度，还可以使用 PixelCNNs 容量捕获全局结构而不是图像的低级统计数据。

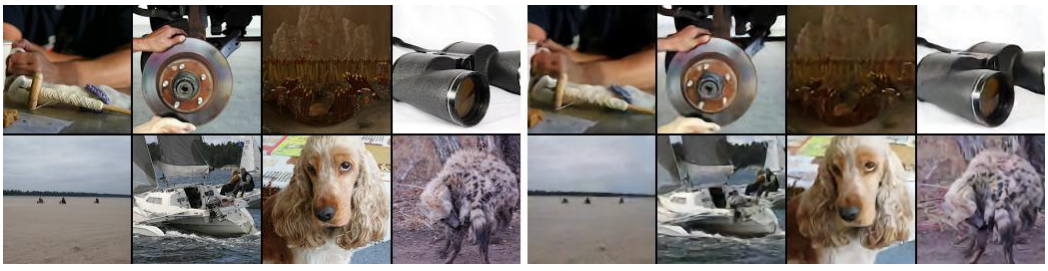

图 2：左：ImageNet 128x128x3 图像，右图：来自 VQ-VAE 的重建，具有 32x32x1 潜在空间，K = 512。

具有离散潜编码的 32x32x1 空间的重建如图 2 所示。即使考虑到我们通过离散编码大大降低了维度，重建看起来只比原始模糊一点。 在这里可以使用比 MSE 更多的感知损失函数（例如，GAN [12]），但我们将其作为未来工作。

接下来，我们在离散的 32x32x1 潜在空间之前训练 PixelCNN。 由于我们只有 1 个通道（不像颜色那样是 3 个），我们只需要在 PixelCNN 中使用空间屏蔽。 我们使用的 PixelCNN 的容量类似于 PixelCNN 论文作者使用的容量[38]。

图 3：来自 VQ-VAE 的样本（128x128），其中 PixelCNN 在 ImageNet 图像上经过训练。 从左到右：套装狐狸，灰鲸，棕熊，海军上将（蝴蝶），珊瑚礁，阿尔卑斯山脉，微波炉，皮卡。

使用 VQ-VAE 的解码器将从 PixelCNN 绘制的样本映射到像素空间，如图 3 所示。

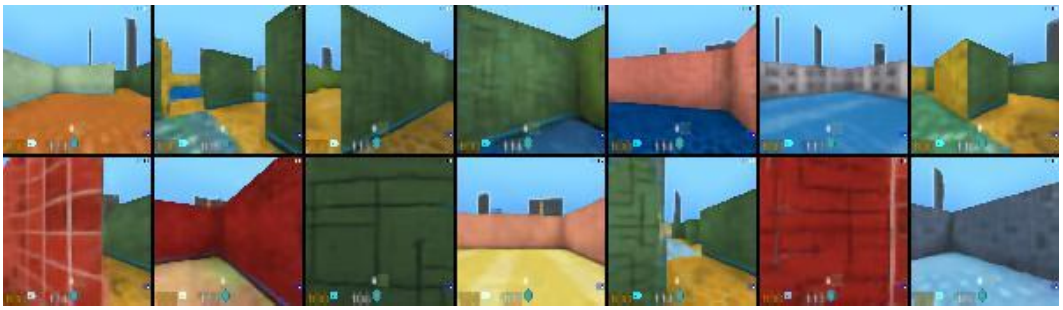

图 4：来自 VQ-VAE 的样品（128x128），其中 PixelCNN 先前已在 DeepMind Lab 捕获的帧上进行了训练。

我们还对从 DeepMind Lab 环境[2]中提取的 84x84x3 帧重复相同的实验。重建看起来几乎与他们的原件相同。在 PixelCNN 之前从 21x21x1 潜在空间进行训练并使用去卷积模型解码器解码到像素空间的样本可以在图 4 中看到。

最后，我们在 DM-LAB 帧上的第一个 VQ-VAE 的 21x21x1 潜在空间之上训练第二个带有 PixelCNN 解码器的 VQ-VAE。这种设置通常会破坏 VAE，因为它们遭受“后验坍塌”，即由于解码器足够强大以完美地模拟 x，因此忽略了潜编码。然而，我们的模型并没有受此影响，并且潜编码被有意义地使用。我们在第二阶段仅使用三个潜在变量（每个都有 K = 512 和它们自己的嵌入空间 e）来对整个图像进行建模，因此模型无法完美地重建图像 - 这是将图像压缩到 3 x 9 位上的结果，即小于 float32。从离散的全局代码中抽样的重建可以在图 5 中看到。

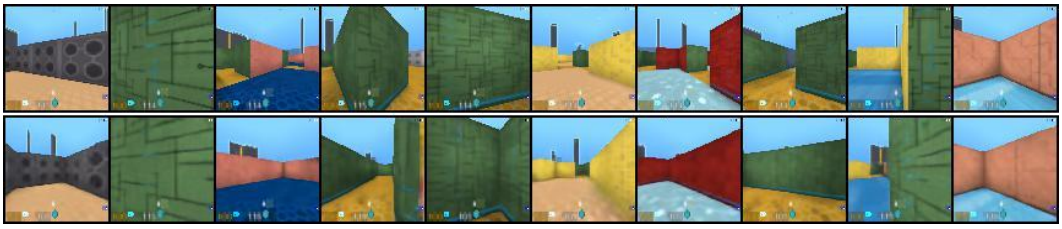

图 5：顶部为原始图像，底部：来自 2 级 VQ-VAE 的重建，具有 3 个潜编码变量以模拟整个图像（27 位），因此模型不能完美地重建图像。 通过在第一 VQ-VAE 的 21x21 潜在域中之前从第二 PixelCNN 采样来生成重建，然后使用标准 VQ-VAE 解码器将其解码为 84x84。 很多原始场景，包括纹理，房间布局和附近的墙壁仍然存在，但模型不会尝试存储像素值本身，这意味着纹理是由 PixelCNN 程序生成的。

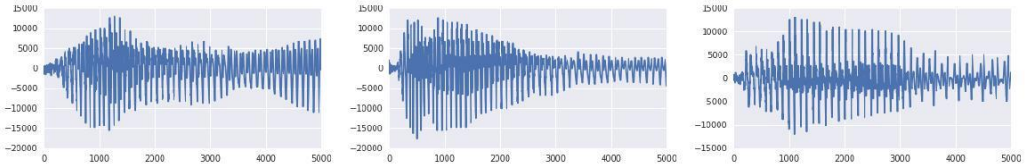

图 6：左：原始波形，中间：使用相同的扬声器 ID 重建，右：使用不同的扬声器 ID 重建。 三个波形的内容相同。

4.3 音频

在这组实验中，我们评估离散潜编码变量在原始音频模型上的行为。在我们所有的音频实验中，我们训练了一个 VQ-VAE，它具有类似于 WaveNet 解码器的扩张卷积结构。本节的所有示例均可通过以下网址进行播放：https://avdnoord.github.io/homepage/vqvae/。

我们首先考虑 VCTK 数据集，其中包含 109 个不同发言者的语音记录[41]。我们训练一个 VQ-VAE，其中编码器有 6 个步幅，步幅为 2，窗口大小为 4。这产生的潜在空间比原始波形小 64 倍。编码点由一个特征图组成，离散空间为 512 维。解码器以隐藏和扬声器的 one-hot 嵌入为条件。

首先，我们进行了一项实验，以证明 VQ-VAE 可以提取仅保留长期相关信息的潜在空间。在训练模型之后，给出音频示例，我们可以将其编码为离散潜在表示，并通过从解码器采样来重建。由于离散表示的维数小 64 倍，原始样本无法逐个样本地完美重建。从所提供的样本中可以听到，并且如图 7 所示，重建具有相同的内容（相同的文本内容），但是波形是完全不同的并且声音中的韵律被改变。这意味着 VQ-VAE 在没有任何形式的语言监督的情况下，学习了一个高级抽象空间，该空间对低级特征不变，只对语音内容进行编码。这个实验证实了我们之前的观察结果，即重要特征通常是那些跨越输入数据空间中许多维度的特征（在这种情况下是音素和波形中的其他高级内容）。

然后，我们分析了模型中的无条件样本，以了解其功能。给定从音频中提取的紧凑和抽象的潜在表示，我们在该表示的顶部训练先验以模拟数据中的长期依赖性。为此，我们使用了 460 个扬声器[30]的更大数据集，并训练了 VQ-VAE 模型，其中离散空间的分辨率小了 128 倍。接下来，我们像往常一样在 40960 时间步长（2.56 秒）的块上对此表示进行训练，产生 320 个潜在的时间步长。虽然从最好的语音模型（如原始的 WaveNet [37]）中抽取的样本听起来像是咿呀学语，但来自 VQ-VAE 的样本包含清晰的单词和部分句子（参见上面链接的样本）。我们得出结论，VQ-VAE 能够以完全无监督的方式从原始音频波形模拟基本的音素级语言模型。

接下来，我们尝试了扬声器转换，其中从一个扬声器提取潜编码，然后使用单独的扬声器 ID 通过解码器重建。从样本中可以听到，合成语音具有与原始样本相同的内容，但具有来自第二发言者的语音。该实验再次证明编码表示已经考虑了特定于说话者的信息：嵌入不仅具有与波形中的细节相同的含义，而且还具有不同的语音特征。

最后，为了更好地理解离散编码的内容，我们将离散编码与完全真实音素序列一对一地进行了比较（没有以任何方式训练 VQ-VAE）。使用以 25 Hz 运行的 128 维离散空间（编码器下采样因子为 640），我们将 128 个可能的潜在值中的每一个映射到 41 个可能的音素值（见底部 1）中的一个（通过有条件地最有可能的音素）。这种 41 路分类的准确度为 49.3%，而随机潜在空间将导致 7.2% 的准确度（之前最可能的音素）。很明显，以完全无监督的方式获得的这些离散潜编码是与音素密切相关的高级语音描述符。

4.4 视频

对于我们的最终实验，我们使用 DeepMind Lab [2]环境来训练以给定动作序列为条件的生成模型。在图 7 中，我们显示了输入到模型的最初 6 帧，然后是从 VQ-VAE 采样的 10 帧，所有操作都设置为向前（顶行）和右（底行）。使用 VQ-VAE 模型生成视频序列完全在潜在空间 z_t 中完成，而无需自己生成实际图像。然后，在仅使用先验模型 $p(z_1, \ldots, z_T)$ 生成所有潜编码之后，通过将具有确定性解码器的潜在编码映射到像素空间来创建序列 x_t 中的每个图像。因此，VQ-VAE 可用于想象纯粹在潜在空间中的长序列而不依赖于像素空间。可以看出，该模型已经学会了成功地生成以给定动作为条件的帧序列，而视觉质量没有任何降低，同时保持局部几何形状正确。为了完整起见，我们训练了一个没有动作的模型并获得了类似的结果，由于空间限制而未显示。

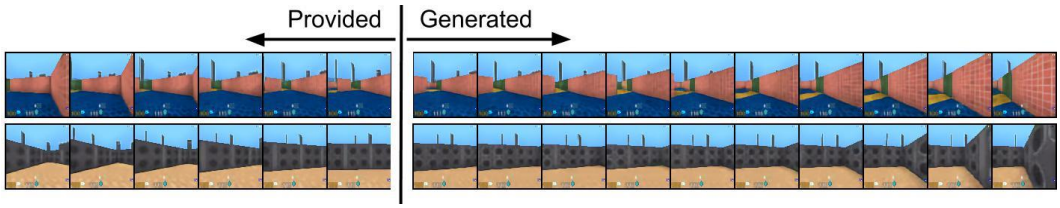

图 7：向模型提供前 6 帧，后续帧以动作为条件生成。 上：重复动作“前进”，下方：重复动作“向右移动”。

5 结论

在这项工作中，我们引入了 VQ-VAE，这是一个新的模型系列，它将 VAE 与矢量量化相结合，以获得离散的潜在表示。我们已经证明 VQ-VAE 能够通过其压缩的离散潜在空间建模非常长期的依赖关系，我们通过生成 128×128 的彩色图像，采样动作条件视频序列以及最终使用音频来证明甚至无条件模型可以产生令人惊讶的有意义的演讲和演讲者转换。所有这些实验表明，VQ-VAE 学习的离散潜在空间以完全无监督的方式捕获数据的重要特征。此外，VQ-VAE 的似然度几乎与 CIFAR10 数据上的连续潜在变量对应物一样好。我们相信这是第一个能成功建模长程序列的离散潜变量模型，并且完全无监督地学习与音素密切相关的高级语音描述符。

[1] 注意，编码器/解码器对可以使每个离散潜在的含义取决于序列中的先前编码，例如 bi / tri-gram（从而实现更高的压缩），这意味着更高级的到音素的映射将导致更高的准确性。

参考文献

[1] Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, and Luc Van Gool. Soft-to-hard vector quantization for end-to-end learned compression of images and neural networks. arXiv preprint arXiv:1704.00648, 2017.

[2] Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. arXiv preprint arXiv:1612.03801, 2016.

[3] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432, 2013.

[4] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349, 2015.

[5] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. arXiv preprint arXiv:1509.00519, 2015.

[6] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. CoRR, abs/1606.03657, 2016.

[7] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. arXiv preprint arXiv:1611.02731, 2016.

[8] Aaron Courville, James Bergstra, and Yoshua Bengio. A spike and slab restricted boltzmann machine. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pages 233–241, 2011.

[9] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. arXiv preprint arXiv:1611.06430, 2016.

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016.

[11] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In Advances in Neural Information Processing Systems, pages 64–72, 2016.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[13] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In Advances In Neural Information Processing Systems, pages 3549–3557, 2016.

[14] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. arXiv preprint arXiv:1310.8499, 2013.

[15] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vázquez, and Aaron C. Courville. Pixelvae: A latent variable model for natural images. CoRR, abs/1611.05013, 2016.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.

[17] Judy Hoffman, Erik Rodner, Jeff Donahue, Trevor Darrell, and Kate Saenko. Efficient learning of domain-invariant image representations. arXiv preprint arXiv:1301.3224, 2013.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004, 2016.

[19] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144, 2016.

[20] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. arXiv preprint arXiv:1610.00527, 2016.

[21] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[22] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. NIPS 2016, 2016.

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802, 2016.

[25] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712, 2016.

[26] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. arXiv preprint arXiv:1612.07837, 2016.

[27] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. arXiv preprint arXiv:1402.0030, 2014.

[28] Andriy Mnih and Danilo Jimenez Rezende. Variational inference for monte carlo objectives. CoRR, abs/1602.06725, 2016.

[29] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759, 2016.

[30] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pages 5206–5210. IEEE, 2015.

[31] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770, 2015.

[32] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approxi-mate inference in deep generative models. arXiv preprint arXiv:1401.4082, 2014.

[33] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In Artificial Intelligence and Statistics, pages 448–455, 2009.

[34] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065, 2016.

[35] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction, volume 1. MIT press Cambridge, 1998.

[36] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. arXiv preprint arXiv:1703.00395, 2017.

[37] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. CoRR abs/1609.03499, 2016.

[38] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In Advances in Neural Information Processing Systems, pages 4790–4798, 2016.

[39] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec):3371–3408, 2010.

[40] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3156–3164, 2015.

[41] Junichi Yamagishi. English multi-speaker corpus for cstr voice cloning toolkit. URL http://homepages. inf. ed. ac. uk/jyamagis/page3/page58/page58. html, 2012.

[42] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. CoRR, abs/1702.08139, 2017.

A 附录

A.1 VQ-VAE 更新字典使用指数移动平均值

如 3.2 节所述，还可以使用指数移动平均值（EMA）来更新字典项而不是公式 3 中的损失项：

$$\|\text{sg}[z_e(x)] - e\|_2^2. \tag{4}$$

让 $\{z_{i,1}, z_{i,2}, \ldots, z_{i,n_i}\}$ 是来自编码器的最接近字典项 e_i 的 n_i 输出的集合，因此我们可以将损失写为：

$$\sum_j^{n_i} \|z_{i,j} - e_i\|_2^2. \tag{5}$$

e_i 的最佳值有一个封闭形式的解决方案，它只是集合中元素的平均值：

$$e_i = \frac{1}{n_i} \sum_j^{n_i} z_{i,j}.$$

此更新通常用于诸如 K-Means 之类的算法中。

但是，在使用 mini-batches 时，我们无法直接使用此更新。 相反，我们可以使用指数移动平均值作为此更新的在线版本：

$$N_i^{(t)} := N_i^{(t-1)} * \gamma + n_i^{(t)}(1-\gamma) \tag{6}$$

$$m_i^{(t)} := m_i^{(t-1)} * \gamma + \sum_j z_{i,j}^{(t)}(1-\gamma) \tag{7}$$

$$e_i^{(t)} := \frac{m_i^{(t)}}{N_i^{(t)}}, \tag{8}$$

其中γ值介于 0 和 1 之间。我们发现γ = 0.99 在实践中运作良好。